

$$y(j) = u(j)(1 + \cos(2\pi j/256))/2$$

for the variable j presumed translated into the range -128 to +127. Do this windowing before the scaling to help avoid overflow on the much larger than average samples as they could fail at the edges of the window. Of course, this windowing could follow the scaling of the next step.

- (5) Scale the windowed input samples simply by left shifting $(2N+8-S)/2-H$ bits (if the number of bits is negative, then this is a right shift). If a sample has magnitude more than 2^H times the average, then overflow will occur and in this case just replace the scaled sample with the corresponding maximum magnitude (e.g., 8000 or 7FFF). Indeed, if the sign bit changes, then overflow has occurred and the scaled sample is taken as the corresponding maximum magnitude. Thus with $y_s(j)$ denoting the scaled windowed samples and no overflow:

$$y_s(j) = y(j)2^{(2N+8-S)/2-H}$$

- (6) Compute the FFT using $y_s(j)$ to find $Y_s(\omega)$. The use of $y_s(j)$ avoids the loss of precision which otherwise would have occurred with the FFT due to underflow avoidance.
- (7) Apply a local smoothing window to $Y_s(\omega)$ as in step (3) of the spectral subtraction preferred embodiments.
- (8) Scale down by shifting $Y_s(\omega)$ $(2N+8-S)/2-H$ bits to the right (with the new sign bit repeating the original sign bit) to have $Y(\omega)$ for noise estimation and filter application in the preferred embodiments previously described.

An alternative precision control scaling uses the sum of the absolute values of the samples in a frame rather than the power estimate (sum of the squares of the samples). As with the power estimate scaling, count the number S of significant bits is the sum of absolute values and scale the input samples by a factor of $2^{N+8-S-H}$ where again $N+1$ is the number bits in the sample representation, the 8 comes from the 256 (2^8) sample frame size, and H provides headroom bits. Heuristically, with samples of K significant bits on the average, the sum of absolute values should be about $K+8$ bits, and so S will be about $K+8$ and the factor will be 2^{N-K-H} which is the same as the power estimate sum scaling.

Further, even using the power estimate sum with S significant bits, scaling factors such as $2^{(2N+8-S)/2-H}$ have yielded good results. That is, variations of the method of scaling up according to a frame characteristic, processing, and then scaling down will also be viable provided the scaling does not lead to excessive overflow.

FIG. 13 illustrates in block format a internal precision controller preferred embodiment which could be used with any of the foregoing noise suppression filter preferred embodiments. In particular, frame energy measurer 1302 determines the scaling factor to be used, and scaler 1304 applies the scaling factor to the input frame. Filter 1306

filters the scaled frame, and inverse scaler 1308 then undoes the scaling to return to the original input signal levels. Filter 1306 could be any of the foregoing preferred embodiment filters. Parameters from filter 1306 may be part of the scale factor determination by measurer 1302. And insertion of noise suppressors 1300 into the systems of FIGS. 1a-b provides preferred embodiment systems in which noise suppressor 1300 in part controls the output.

Modifications

The preferred embodiments may be varied in many ways while retaining one or more of the features of clamping, noise enhancing, smoothed power estimating, recursive noise estimating, adaptive clamping, adaptive noise suppression factoring, codebook based estimating, and internal precision controlling.

For example, the various generalized Wiener filters of the preferred embodiments had power β equal to $1/2$, but other powers such as $1/4$, $1/8$, and so forth also apply; higher filter powers imply stronger filtering. The frame size of 256 samples could be increased or decreased, although powers of 2 are convenient for FFTs. The particular choice of 3 bits of additional headroom could be varied, especially with different size frames and different number of bits in the sample representation. The adaptive clamp could have a negative dependence upon frame noise and signal estimates ($B < 0$). Also, the adaptive clamp could invoke a near-end speech detection method to adjust the clamp level. The α and κ coefficients could be varied and could enter the transfer functions as simple analytic functions of the ratios, and the number iterations in the codebook based generalized Wiener filter could be varied.

What is claimed is:

1. A method of filtering a stream of sampled acoustic signals, comprising the steps of:

- (a) partitioning a stream of sampled acoustic signals into a sequence of frames;
- (b) Fourier transforming said frames to yield a sequence of transformed frames;
- (c) applying a generalized Wiener filter to said transformed frames to yield a sequence of filtered transformed frames, wherein said filter uses power spectrum estimates from LSFs defined as weighted sums of LSFs of a codebook of LSFs with the weights determined by the LSFs of said transformed frames; and
- (d) inverse Fourier transforming said sequence of filtered transformed frames to yield a sequence of filtered frames.

2. The method of claim 1, further comprising the steps of:

- (a) repeating step (c) of claim 1 but with the LSFs of said transformed frame replaced with the LSFs of the filtered transformed frame of a preceding iteration of said step (c) of claim 1.

3. The method of claim 2, wherein:

- (a) said step (c) of claim 1 is repeated a number of times with the number in the range of 6 to 7.

* * * * *